University of Zurich[UZH]

Zurich Open Repository and Archive
University of Zurich
University Library
Strickhofstrasse 39
CH-8057 Zurich
www.zora.uzh.ch

Year: 2018

UZH@SMM4H: System Descriptions

Ellendorff, Tilia ; Cornelius, Joseph ; Gordon, Heath ; Colic, Nicola ; Rinaldi, Fabio

Abstract: Our team at the University of Zurich participated in the first 3 of the 4 sub-tasks at the Social Media Mining for Health Applications (SMM4H) shared task. We experimented with different approaches for text classification, namely traditional feature-based classifiers (Logistic Regression and Support Vector Machines), shallow neural networks, RCNNs, and CNNs. This system description paper provides details regarding the different system architectures and the achieved results.

Posted at the Zurich Open Repository and Archive, University of Zurich
ZORA URL: https://doi.org/10.5167/uzh-168481
Conference or Workshop Item
Published Version

Originally published at:
Ellendorff, Tilia; Cornelius, Joseph; Gordon, Heath; Colic, Nicola; Rinaldi, Fabio (2018). UZH@SMM4H: System Descriptions. In: SMM4H: The 3rd Social Media Mining for Health Applications Workshop and Shared Task, Brussels, 30 November 2018. Association for Computational Linguistics, 56-60.

UZH@SMM4H: System Descriptions

Tilia Ellendorff[*], **Joseph Cornelius**[†], **Heath Gordon**[*], **Nicola Colic**[*], **Fabio Rinaldi**[*]
[*]Institute of Computational Linguistics, University of Zurich ({name.surname}@uzh.ch)
[†]Institute of Neuroinformatics, University of Zurich and ETH (jocorn@ini.phys.ethz.ch)

Abstract

Our team at the University of Zürich participated in the first 3 of the 4 sub-tasks at the Social Media Mining for Health Applications (SMM4H) shared task. We experimented with different approaches for text classification, namely traditional feature-based classifiers (Logistic Regression and Support Vector Machines), shallow neural networks, RCNNs, and CNNs. This system description paper provides details regarding the different system architectures and the achieved results.

1 Introduction

The 2018 edition of the Social Media Mining for Health Applications (SMM4H) challenge (Weissenbacher et al., 2018) consists of 4 tasks, all of which can be framed as document classification problems: automatic detection of posts mentioning a drug name (task 1), automatic classification of posts describing medication intake (task 2), automatic classification of posts mentioning adverse drug reaction (task 3) and vaccination behavior, respectively (task 4). Our team participated in the first three of them.

While tasks 1 (*drug name detection*) and 3 (*adverse drug reaction mentioning detection*; ADR) consisted in binary text classification, task 2 (*medication intake classification*) included three classes: *personal medication intake*, *possible medication intake* and *non-intake*.

2 Data Description and Pre-processing

For each task, participants were provided with a dataset. The tweets were provided by ID, and had to be downloaded individually from Twitter. Because of that, not all tweets in the datasets were available anymore at the time of our participation. The number of tweets that we had at our disposal are shown in Table 1.

Task	Labels	Tweets	Train	Develop.
1	0	4462	4357	105
	1	4776	4657	119
Total		9238	9014	224
2	1	3198	3129	69
	2	5162	5058	104
	3	7155	7028	127
Total		15515	15215	300
3	0	15416	15148	268
	1	1359	1327	32
Total		16775	16475	300

Table 1: Overview of available tweets for each task, split by us into a training and a small development set.

We tokenized the tweets using SpaCy (Honnibal and Montani, 2017) and applied a number of pre-processing steps before further processing the tweets. These include the following:

- We found that URLs are frequently merged with the preceding token. Therefore, we split before URLs (i.e. before "http").
- We split URLs into their components (i.e. parts of each URL are treated as separate tokens).
- We split all tokens at camel case (e.g. "MedicationProblems" is split into "Medication" and "Problems").
- We stripped the hashtag symbol (#) from all tokens where it applies.
- We replaced "w/" and "w/o" by their full versions ("with" and "without").
- We additionally split at the following punctuation symbols: -/.]
- We replaced numbers and usernames by placeholder tokens.

Following these pre-processing steps, we used SpaCy for lemmatization and part-of-speech tagging.

Proceedings of the 3rd Social Media Mining for Health Applications (SMM4H) Workshop & Shared Task, pages 56–60
Brussels, Belgium, October 31, 2018. ©2018 Association for Computational Linguistics

	Task 1			Task 2			Task 3		
	P	R	F	P	R	F	P	R	F
Logreg:	0.861	0.861	0.861	0.591	0.565	0.578	0.917	0.344	0.500
MLP:	0.861	0.861	0.861	0.679	0.522	0.590	0.750	0.281	0.409
Lin SVM:	0.534	0.534	0.534	0.575	0.609	0.592	0.571	0.375	0.453
Shallow NN:	0.577	0.381	0.459	0.780	0.565	0.655	0.550	0.344	0.423
RCNN:	0.916	0.924	0.920	0.468	0.638	0.540	0.185	0.156	0.169

Table 2: Results of feature based systems on the development set. For task 2, results are micro-averaged over the two positive labels only. For task 3, they are micro-averaged over the positive label only.

3 System Descriptions

The following sections give an overview of the system architectures with which we experimented. The results obtained on our development set by a selection of configurations (as described below) are shown in Table 2.

3.1 Feature-based systems

The feature-based classifiers include the following features:

- bag-of-lemmas (unigrams and bigrams)
- averaged pre-trained word embeddings (Sarker and Gonzalez, 2017)
- a binary feature providing information if the tweet contains any token found in a terminology list described later in this section
- a set of features recording all exact matches from that list as observed in the training data. This allows the classifier to assign a weight depending on the number of positive or negative tweets in which these terms are observed.

We experimented with Naive Bayes, linear SVMs, Logistic Regression and Multilayer Perceptron classifiers. However, since Naive Bayes consistently gave us the worst performance on the development set, we excluded it. Our Multilayer Perceptron has two hidden layers using tanh activation with 100 and 50 units, respectively, and applies an Adam optimizer with an adaptive learning rate.

To improve on our results, we employ two term lists; one with terms derived from an external resource, and one with terms extracted from the task data. Firstly, we use a manually curated list of drug names, derived from RX Norm (Nelson et al., 2011), which we had originally created for a different project. RX Norm is a normalized list of all clinical drugs available in the US, indexing them by commercial name and compounds. We compared this list with a list of the 10000 most common English words in order to determine the amount of ambiguity and found only a negligible overlap. This means that both chemical compounds and brand names are very specific in most cases and therefore only show a very small amount of ambiguity.

142	drug names
85	chemical compounds
42	class of drugs (such as analgesics)
41	misspellings
39	symptoms
38	related term (such as addictive)
17	hashtag (such as #advilsinuscrowd)
16	abbreviations (such as alka)
12	pharmaceutical company
73	others (plants, medical devices etc)
505	**total**

Table 3: Categories of terms derived from tweets.

However, to better account for the fact that social media data is noisy, and users misspell and abbreviate drug names or use different names not contained in the vocabulary, we constructed a second list: We gathered tokens from positive tweets in the training data which do not contain any drug names from the list above. From this set, only tokens that do not occur in the 20000 most frequent natural language words as computed on Google Books Ngram Corpus were kept, and evaluated manually if they refer to a drug. This method revealed common misspellings such as `adderal` (instead of `adderall`) or `codrol` (instead of `codral`), but also lead to the identification of several word categories that can be positive predictors for drug usage, such as diseases and symptoms. Table 3 lists the categories of the terms extracted in this fashion.

3.2 Shallow Neural Network with Tunable Embeddings

This is a simple system based on end-to-end learning within a shallow neural network. The first layer consists of tunable pre-trained embeddings followed by average pooling and a dense layer with sigmoid activation to reach a final classification decision. The embedding layer uses the fasttext embeddings trained on the English version of Wikipedia (Bojanowski et al., 2016), which, during training, we fine-tune to the task. We use cross-entropy as loss function and Adam for optimization. Furthermore, we apply early stopping using a small portion of the training set.

3.3 Recurrent Convolutional Neural Network (RCNN)

For task 1, we additionally apply a combination of a recurrent neural network and a convolutional neural network. The recurrent convolutional neural network (RCNN) (Lai et al., 2015) uses recurrent structures which enables it to capture the context information of each word while simultaneously producing minimal noise. Additionally, it uses a max-pooling layer to capture the relevance of every word in the text. Our recurrent structure is a two layer stacked bidirectional network with gated recurrent unit (GRU) (Cho et al., 2014) cells. The final hidden states of the recurrent structure are the input to the 1D-max-pooling layer. The model is based on Prakash Pandey's implementation of Text Classification in PyTorch. We experimented with a character-based and a word-based version, which are described in the following.

The words of the lemmatized version of the tweets serve as input for the word-based RCNN. We have experimented with various word embeddings in combination with and without a lemmatized input. We used embeddings that were trained on a lemmatized corpus of tweets as well as embeddings that were trained on a non-lemmatized corpus of tweets. In the final version we used embeddings trained on a non-lemmatized corpus for processing the lemmatized version of corpus. Even if it might seems methodologically incorrect, it this was relatively simple to do, and it was empirically found to produce the best results on our development corpus. Our model has 256 hidden states for each direction of the bidirectional recurrent network. We use Adam with a learning rate of 0.00008 for the optimization and softmax cross entropy to compute the loss. We utilize L2 regularization with a rate of 0.005 and to prevent overfitting, we employ a dropout with a rate of 0.8. The whole system learns for 45 epochs with a batch size of 256. We apply domain-specific, pretrained word embeddings (Sarker and Gonzalez, 2017) that are trained on 1 billion tweets from drug-related conversations on Twitter. In addition, we append one dimension to the word embeddings to determine whether a word is listed in the list of RX Norm drug names mentioned before. If a word is included in the list, we insert the value 5 and if it is not included we insert the value -1. In the case of drug names for which no pre-trained embedding is available, we generate a generic embedding vector by averaging the vectors of all RX terms for which an embedding can be found.

We also experimented with a character-based RCNN (using 1-grams and 3-grams) as another approach to the detection of drug name mentioning tweets. However, we did not use this system in the final submission since its performance was consistently worse than that of the word-based model. We considered a combination of the two approaches, however we could not implement it due to lack of time. As input we use the lemmatized version of the tweets converted to the specific character N-gram. Each N-gram corresponds to a unit that is fed into the RNN in the form of its embedding. The same hyperparameters as in the word-based RCNN are used in the character-based RCNN, except that we apply a learning rate of 0.0001, an L2 regularization with a rate of 0.001 and train the system only for 40 epochs. To test the model we have divided the given training set into a validation, test and training set[1]. With the test set, which contains 1000 samples, we achieve an precision of 0.8879, a recall of 0.8840 and an F1 score of 0.8860.

After the challenge, we performed a small error analysis on the results obtained by the word-based RCNN over the 244 tweets in the development set. We found that 19 tweets were misclassified, 10 tweets were mistakenly classified as drug name mentioning, but six of them contain words like 'vitamin', 'maca' and 'pills'. The distinction made by the dataset between the fine nuances of some substances as endogenous rather than as a supplement seems to be a problem for the system. Nine

[1]The partition of the data used within this set-up is different from the partitions reported in table 1

tweets were falsely identified as drug-free tweets, one of these contains a misspelled drug name and another one contains incorrectly tokenized words (e.g. *aspirin!*). Another two tweets contained the word "weed" in the medical context and the word "cannabinoid", which may have been classified as drug names rather than medication names. For the remaining tweets, there is no specific property that would lead to a misclassification.

For comparison, we briefly report on results obtained using a simple biLSTM (bidirectional long short term memory) model as a baseline. The biLSTM model has 128 hidden states for each direction of the bidirectional recurrent network. Adam is used for the optimization with a learning rate of 0.0001. As loss function we use softmax cross entropy. We apply L2 regularization with a rate of 0.005 and a dropout with a rate of 0.5 to achieve a greater generalization. We run the model for 30 epochs with a batch size of 128. On the development set with 1000 tweets the baseline biLSTM achieves an F1 score of 0.881, while the previously described final version of the RCNN achieves an F1 score of 0.92.

3.4 Ensemble of Convolutional Neural Networks (CNNs)

For task 3 we created a tiered ensemble of convolutional neural networks (CNNs). To create the first ensemble, we downsampled the majority class by splitting it up into 5 equally sized training sets. The minority class data remained unsampled and was paired with each of those majority class splits, giving 5 data sets of a 1:2 minority-majority ratio. A CNN was trained on each of these samples for 20 epochs. Input features are the pre-trained word embeddings by Sarker and Gonzalez (2017). The decisions of each CNN for each sample are fed into a simple voting classifier. Twenty of these ensembles were created, and their predictions are fed into a simple majority vote classifier, forming the final set of predictions. This system is based on the general architecture used by (Friedrichs et al., 2018) whereas the individual CNNs is based on (Kim, 2014). Despite promising results on our development set (F: 48.3), this setting performed poorly in the official evaluation (Run 3 of task 3), probably due to a configuration error.

		Precision	Recall	F-Score
Task 1	run 1	0.908	0.834	0.870
	run 2	0.927	0.878	0.902
	run 3	0.908	0.856	0.878
	Mean	**0.890**	**0.872**	**0.880**
Task 2	run 1	0.315	0.434	0.365
	run 2	0.371	0.437	0.401
	run 3	0.431	0.368	0.397
	Best	**0.654**	**0.783**	**0.713**
Task 3	run 1	0.593	0.231	0.333
	run 2	0.455	0.436	0.445
	run 3	0.132	0.935	0.232
	Best	**0.442**	**0.636**	**0.522**

Table 4: Official scores for our submissions, compared with scores obtained by other participating systems.

4 Results

Our official results are summarized in Table 4, compared with other official results as currently available to us (score means for task 1, scores of the system with best F-score for task 2 and 3). Below a brief description of our submitted runs.

For **task 1**, Run 1 was based on the best performing among the models described in section 3.1 (logistic regression). Run 2 was based on the word-based RCNN model described in 3.3. Run 3 was based on a rather crude attempt to improve the scores of a previous run based on a manually curated version of the drug list described in section 3.1, flipping a negative into a positive if the presence of one of terms in the list was detected. The idea was that such a presence should be considered as a highly reliable indicator that the tweet is positive. Experiments on the training corpus had shown generally an increase in recall with a minimal loss in precision. This was confirmed in the official results, where the correction described above was applied to the results of Run 1. However, after the competition we asked the organizers to score another run generated by applying the same "corrector" to Run 2, but in this case the results improved only slightly (P: 0.890, R: 0.872, F:0.880).

For **task 2**, Run 1 was based on the best performing among the models described in section 3.1 (linear SVM), Run 2 with the second best model (Multilayer Perceptron), Run 3 with the Shallow Neural Network model described in section 3.2.

For **task 3**, Run 1 was generated with the Lo-

gistic Regression model, Run 2 with the Shallow Neural Network model, Run 3 with the complex CNN ensemble approach described in section 3.4. However, we suspect a bug in the latter approach since the results were worse than anticipated.

5 Conclusion

In this system description paper we provide details and results for the different approaches with which we experimented for our participation in 3 sub-tasks of the SMM4H shared task. Our interest in this shared task stems from the fact that we are involved in a recently started research project where we will process social media data, including tweets in a related domain. Therefore we plan to continue our experiments with the datasets provided and report new results at the final workshop.

Acknowledgments

We gratefully acknowledge the support of the Swiss National Science Foundation (grants $CR30I1\_162758$ and 407440_167381) and Innosuisse (grant $25587.2PFES-ES$).

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv preprint arXiv:1607.04606*.

Kyunghyun Cho, Bart van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Jasper Friedrichs, Debanjan Mahata, and Shubham Gupta. 2018. InfyNLP at SMM4H Task 2: Stacked Ensemble of Shallow Convolutional Neural Networks for Identifying Personal Medication Intake from Twitter. *arXiv preprint arXiv:1803.07718*.

Matthew Honnibal and Ines Montani. 2017. SpaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *To appear*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent Convolutional Neural Networks for Text Classification. *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2267–2273.

Stuart J Nelson, Kelly Zeng, John Kilbourne, Tammy Powell, and Robin Moore. 2011. Normalized names for clinical drugs: Rxnorm at 6 years. *Journal of the American Medical Informatics Association*, 18(4):441–448.

Abeed Sarker and Graciela Gonzalez. 2017. A corpus for mining drug-related knowledge from Twitter chatter: Language models and their utilities. *Data in Brief*, 10:122–131.

Davy Weissenbacher, Abeed Sarker, Michael Paul, and Graciela Gonzalez-Hernandez. 2018. Overview of the Third Social Media Mining for Health (SMM4H) Shared Tasks at EMNLP 2018. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.